



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/943,900	08/30/2001	Yan J. Arrouye	P2209USC1	2367

7590

02/09/2006

APPLE COMPUTER, INC.
1 INFINITE LOOP, M/S 3-PAT
CUPERTINO, CA 95014

EXAMINER

BULLOCK JR, LEWIS ALEXANDER

ART UNIT	PAPER NUMBER
----------	--------------

2195

DATE MAILED: 02/09/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/943,900

Applicant(s)

ARROUYE ET AL

Examiner

Lewis A. Bullock, Jr.

Art Unit

2195

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 28 November 2005.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 24-37 and 52-75 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 24-37 and 52-75 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 04 February 2002 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

2. Claims 24-37 and 52-75 are rejected under 35 U.S.C. 103(a) as being unpatentable over CROWSAR (U.S. Patent 5,615,400) in view of THATTE (U.S. patent 4,695,949).

As to claim 27, CROWSAR teaches a computer system for dynamically and automatically loading and unloading a software library (libraries) to and from memory in a computer, the software library including one or more library routines (functions) and capable of being referenced by an application or other software module (col. 4, lines 33-45; col. 4, lines 55-56; col. 13, line 64 – col. 14, line 23), the computer system comprising:

one or more library structures (libraries), each library structure corresponding to a software library, each library structure including a library implementation module (constructor / code resources / implementation segment) containing code for implementing the corresponding software library (col. 9, lines 49 – col. 10, line 18; col. 14, line 65 – col. 14, line 3; col. 15, lines 15-19) and a library loader (jump table) containing entry points corresponding to entry points of the corresponding software library and code for loading and unloading the corresponding library implementation

Art Unit: 2195

module (via the jump table containing pc-relative jsr instructions that goes to the SLM segment loader entry point...) (col. 15, lines 20-61); and software to perform operations comprising: loading into memory a library implementation module (implementation segment) for a software library, said loading occurring prior to when a library routine of the software library is to be executed by the application or other software module (via on-demand loading) (col. 13, line 64 – col. 14, line 6; col. 14, lines 15-23; col. 14, line 65 – col. 14, line 3; col. 15, lines 15-23), and that shared libraries are dynamically loaded and unloaded based on use counts (col. 14, lines 15-23) wherein the load function increments the use count and after access of the library the use count is decremented (col. 11, lines 11-19; col. 9, lines 35-39) such that the memory management system can unload libraries or segments of a library which are not in current use by an active application while maintaining the library loader (col. 4, lines 42-45; col. 56, lines 35-41). It is inherent with the teachings of COWSAR that since the use count is used to determine accessing and loading of a library that if the library is in memory its use-count is incremented and if it is not in memory it is loaded and the use-count is subsequently incremented as well as the library being in memory and not be used, i.e. having a zero use count. However, COWSAR does not explicitly teach if the library implementation module is not in memory and scheduled to be unloaded from memory, then canceling the scheduled unloading of the library implementation module; and automatically scheduling the unloading of the library implementation module; and unless the scheduled unload has been cancelled automatically unloading the library implementation module as scheduled wherein the software library is scheduled for

unload and unloaded automatically without an explicit unload request from the application or software module.

THATTE teaches memory management technique wherein if a referenced module (memory block) is not in memory and scheduled to be unloaded from memory (i.e. the memory block has a zero reference count), then canceling the scheduled unloading of the module (increasing the reference count for the memory block and removing it from the reference count filter) (col. 7, lines 27-39; col. 8, lines 27-38); and automatically scheduling the unloading of the module (via placing the memory block into the reference count filter to be sent to the garbage buffer); and unless the scheduled unload has been cancelled (via a subsequent reference to increase the reference count from zero to one) (col. 7, lines 27-39; col. 8, lines 27-38) automatically unloading the module as scheduled wherein the module is scheduled for unload and unloaded automatically without an explicit unload request from the application or software module (via a background process that garbage the memory blocks that are not referenced or subsequently referenced) (col. 9, lines 1-25). It would be obvious to one skilled in the art at the time of the invention that since the libraries determine access, i.e. reference based upon use counts, that the use counts are the referenced counts and the libraries are the memory blocks based upon the combination. Therefore, it would be obvious to one skilled in the art at the time of the invention to combine the teachings of COWSAR with the teachings of THATTE in order to efficient, cost-effective method of alleviating the need for frequent garbage collection, reduces the reference counting overhead, and allows reference counting to be implemented in practice (col. 5, lines 23-28).

As to claim 24, refer to claim 27 for rejection. However, claim 24 further details determining which libraries are potentially needed during execution of an application; and loading those libraries via the library loader and implementation modules. COWSAR teaches that libraries are on-demand loaded into the system based upon the application, therefore they are referenced and essentially needed by the application (col. 13, line 64 – col. 14, line 6; col. 14, lines 15-23; col. 14, line 65 – col. 14, line 3; col. 15, lines 15-23). COWSAR also teaches loading libraries that have library loaders (jump tables) and implementation modules (implementation segment) (col. 9, lines 49 – col. 10, line 18; col. 14, line 65 – col. 14, line 3; col. 15, lines 15-19) (via the jump table containing pc-relative jsr instructions that goes to the SLM segment loader entry point...) (col. 15, lines 20-61).

As to claim 36, refer to claim 27 for rejection.

As to claims 52-54, refer to claim 27 for rejection. However, claim 52 further details the unloading of the library implementation module includes setting a time delay and unloading the library implementation module if after the time delay has been met the library implementing module is not in use. COWSAR teaches that libraries are referenced by use counts and include implementation modules (col. 9, lines 49 – col. 10, line 18; col. 14, line 65 – col. 14, line 3; col. 15, lines 15-19). THATTE teaches that unreferenced modules are referenced, unreferenced, rereferenced or reclaimed by a

Art Unit: 2195

memory management system when their reference count is zero (col. 8, lines 1-17).

Hence, when an unreferenced module (a memory block having a zero reference count) is re-referenced, the module increments its reference counter from zero to one before it is garbage collected by a background process. Therefore, it would be obvious to one skilled in the art that the garbage collection is performed separately from the handling of the reference counter. "Official Notice" is taken in that it is well known in the art that a user or system has a set time of operation or criteria, i.e. low memory, for performing garbage collection or memory management operations on a computer system and therefore would be obvious in view of COWSAR and THATTE in order to set a time delay or criteria for officially unloading a library, i.e. run the garbage collection algorithm, when the library is not in use.

As to claim 55, refer to claim 52 for rejection.

As to claim 56, refer to claim 52 for rejection.

As to claim 57, refer to claim 52 for rejection.

As to claim 58, refer to claim 52 for rejection.

As to claim 59, refer to claim 24 for rejection.

As to claim 60-62, refer to claim 24 for rejection. However, claim 60 further details the unloading of the library implementation module includes setting a time delay and unloading the library implementation module if after the time delay has been met the library implementing module is not in use. COWSAR teaches that libraries are referenced by use counts and include implementation modules (col. 9, lines 49 – col. 10, line 18; col. 14, line 65 – col. 14, line 3; col. 15, lines 15-19). THATTE teaches that unreferenced modules are referenced, unreferenced, rereferenced or reclaimed by a memory management system when their reference count is zero (col. 8, lines 1-17). Hence, when an unreferenced module (a memory block having a zero reference count) is re-referenced, the module increments its reference counter from zero to one before it is garbage collected by a background process. Therefore, it would be obvious to one skilled in the art that the garbage collection is performed separately from the handling of the reference counter. “Official Notice” is taken in that it is well known in the art that a user or system has a set time of operation or criteria, i.e. low memory, for performing garbage collection or memory management operations on a computer system and therefore would be obvious in view of COWSAR and THATTE in order to set a time delay or criteria for officially unloading a library, i.e. run the garbage collection algorithm, when the library is not in use.

As to claim 63, refer to claim 24 for rejection.

As to claims 64-67, refer to claims 60-62 for rejection.

As to claims 68-73, refer to claims 60-62 for rejection. However, claim 68 further details the system comprising a processor; a memory; and a disk. COWSAR teaches the disclosed limitations (col. 6, lines 19-33).

As to claims 74, refer to claim 60 for rejection.

As to claim 75, refer to claim 60 for rejection.

As to claims 25 and 26, COWSAR teaches loading of library is performed after execution of the application begins but before the library routine is executed (on-demand loading) (libraries are loaded at any time) (col. 13, line 64 – col. 14, line 23).

As to claims 28 and 29, refer to claims 25 and 26 for rejection.

As to claims 30-35, COWSAR teaches that libraries are referenced by use counts and include implementation modules (col. 9, lines 49 – col. 10, line 18; col. 14, line 65 – col. 14, line 3; col. 15, lines 15-19). THATTE teaches that unreferenced modules are referenced, unreferenced, rereferenced or reclaimed by a memory management system when their reference count is zero (col. 8, lines 1-17). Hence, when an unreferenced module (a memory block having a zero reference count) is re-referenced, the module increments its reference counter from zero to one before it is

garbage collected by a background process. Therefore, it would be obvious to one skilled in the art that the garbage collection is performed separately from the handling of the reference counter. "Official Notice" is taken in that it is well known in the art that a user or system has a set time of operation or criteria, i.e. low memory, for performing garbage collection or memory management operations on a computer system and therefore would be obvious in view of COWSAR and THATTE in order to set a time delay or criteria for officially unloading a library, i.e. run the garbage collection algorithm, when the library is not in use.

As to claim 37, refer to claims 29-35 for rejection.

Response to Arguments

3. Applicant's arguments filed November 28, 2005 have been fully considered but they are not persuasive. Applicant argues that Cowser does not teach the limitation of "unloading the library implementation module while keeping the library loader in memory until the execution of the application or other software module is completed" by referring stating that Cowser teaches, "when segments are unloaded the pc relative jsr is put back in the jump table for each entry for that segment. This would imply that the jump table is removed from memory and not maintained as Applicant's claims no state. The examiner disagrees. Cowser also teaches a preload flag that indicates that all segments of the shared library should be loaded at library load time wherein it is not guaranteed that the segments will not be unloaded so the jump table must be kept in

Art Unit: 2195

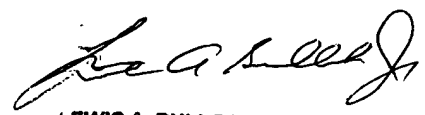
memory and intersegment references are left pointing to the jump table (col. 56, lines 35-41). This would allude to the segments being capable of unloading and the jump table remaining in memory as outlined in the claims. Therefore, the rejection is maintained as disclosed above.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Lewis A. Bullock, Jr. whose telephone number is (571) 272-3759. The examiner can normally be reached on Monday-Friday, 8:30 a.m. - 5:00 p.m..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).


LEWIS A. BULLOCK, JR.
PRIMARY EXAMINER

February 6, 2006